

**Technical Reports on Mathematical and Computing Sciences: TR-C123****title:** Practical Algorithms for On-line Sampling**authors:** Carlos Domingo<sup>1</sup>, Ricard Gavalda<sup>1</sup>, and Osamu Watanabe<sup>2</sup>**affiliation:**

1. Dept. de LSI, Univ. Politecnica de Catalunya  
Campus Nord, Modul C5, 08034-Barcelona, Spain.  
**email:** {carlos, gavalda}@lsi.upc.es

2. Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology  
Meguro-ku Ookayama, Tokyo 152-8552.  
**email:** watanabe@is.titech.ac.jp

**acknowledgements to financial supports:**

1. Supported in part by ESPRIT LTR Project no. 20244 - ALCOM-IT  
and CICYT Project TIC97-1475-CE.
2. Supported in part by the Ministry of Education, Scinece, Sports and Culture,  
Grant-in-Aid for Scientific Research on Priority Areas (Discovery Science), 1998.

**Abstract.** One of the core applications of machine learning to knowledge discovery consists on building a function (a hypothesis) from a given amount of data (for instance a decision tree or a neural network) such that we can use it afterwards to predict new instances of the data. In this paper, we focus on a particular situation where we assume that the hypothesis we want to use for prediction is very simple, and thus, the hypotheses class is of feasible size. We study the problem of how to determine which of the hypotheses in the class is almost the best one. We present two on-line sampling algorithms for selecting hypotheses, give theoretical bounds on the number of necessary examples, and analize them exprimentally. We compare them with the simple batch sampling approach commonly used and show that in most of the situations our algorithms use much fewer number of examples.

## 1 Introduction and Motivation

The ubiquity of computers in business and commerce has lead to generation of huge quantities of stored data. A simple commercial transaction, phone call or use of a credit card is usually stored in a computer. Todays databases are growing in size and therefore there is a clear need for automatic tools for analyzing and understanding these data. The field known as knowledge discovery and data mining aims at understandings and developing all the issues concern with the extraction of patterns from vast amount of data. Some of the techniques used are basically machine learning techniques. However, due to

the restriction that the data available is very large, many machine learning techniques do not always scale well and can not just simply be applied.

One of the core applications of machine learning to knowledge discovery consists of building a function from a given amount data (for instance a decision tree or a neural network) such that we can later use it to predict the behavior of new instances of the data. This is commonly known as concept learning or supervised learning.

Most of the previous research in machine learning has focused on developing efficient techniques for obtaining *highly* accurate predictors. For achieving high accuracy, it is better that learning algorithms can handle complicated predictors, and developing efficient algorithms for complicated predictors has been studied intensively in machine learning.

On the other hand, for knowledge discovery, there are some other aspects of concept learning that should be considered, and we discuss, in this paper, one of them. We study concept learning (or, more simply, hypotheses selection) for a particular situation that we describe in the following. We assume that in our situation we have a class  $\mathcal{H}$  of very simple hypotheses, and we want to select one of the reasonably accurate hypotheses from them, by using a given set of data, i.e., labeled examples. Since hypotheses we deal with are very simple, we cannot hope, in general, to find highly accurate hypotheses in  $\mathcal{H}$ . On the other hand, the size of hypotheses space  $\mathcal{H}$  is relatively small and feasible. We also assume that the size of the data available is huge, and thus, it is very inefficient to use all examples in the dataset. Simple hypotheses have been studied before by several researchers and it has been reported that in some cases they can achieve surprisingly high accuracy (see, e.g., [9, 5, 1]). Moreover, with the new discovery of voting methods like boosting [4], bagging [2], or error-correcting output codes [3], several of these hypotheses can be combined in a way that the overall precision becomes extremely high.

Perhaps the paper by Holte [5] best exemplifies our problem. In that paper he performs several experiments with some datasets from the repository of the University of California at Irvine. His learning algorithm is extremely simple, just obtains a training set from the datasets, it builds a set of very simple hypotheses according to the different features of the dataset (see the paper for more details on how to build the set of simple hypotheses) and then selects the hypothesis that has the highest accuracy on the training set. It turns out that this simple approach is indeed efficient since for most of the datasets the accuracy is between 80 and 90 percent. His choice of training set size is totally arbitrary,  $2/3$  of the whole dataset. If the dataset available is huge as it happens in many situations then this choice might be very inefficient.

On the other hand, the obvious approach for solving this problem that is commonly used in computational learning theory [8] is to first choose randomly a certain number  $m$  of examples from the dataset, and then select the hypothesis that performs best on

these examples. (We will call this simple hypotheses selection *Batch Selection* (BS) in this paper.) The number  $m$  is calculated so that the best hypotheses on the selected sample is close to the real best one with high probability; such  $m$  can be calculated by using uniform convergence bounds like the Chernoff or the Hoeffding bound (see, e.g., [6] for some examples of this approach). However, if we want to apply this method in a real setting we will encounter two problems. First, the theoretical bounds are usually too pessimistic and thus the bounds obtained are not practical. Second, to obtain these bounds we need to have certain knowledge about the accuracy of hypotheses in a given hypothesis space. What is usually assumed is that we know a lower bound on the accuracy of the best hypothesis. Again, this lower bound might be far from the real accuracy of the best hypothesis and thus the theoretical bound becomes too pessimistic. Or even worst, in many applications we just do not know anything about the accuracy of the hypotheses.

In this paper we propose two algorithms for solving this problem, obtain theoretical bounds of their performance, and evaluate them experimentally. Our goal is to obtain algorithms that are useful in practice but that also have certain theoretical guarantees about their performance. The first distinct characteristic is that we obtain the examples in an on-line manner rather than in batch. The second is that the number of examples has less dependency on the lower bound of the accuracy than the above obvious Batch Selection. More specifically, if  $\gamma_0$  is the accuracy of the best hypothesis, and  $\gamma$  is the lower bound for  $\gamma_0$  we would use, then the sample size  $m$  for Batch Selection given by the theoretical bound is  $\mathcal{O}(1/\gamma^2)$  (ignoring dependencies in other parameters). On the other hand, the sample size of our first algorithm is  $\mathcal{O}(1/\gamma\gamma_0)$ , and that of the second one is  $\mathcal{O}(1/\gamma_0^2)$ .

The paper is organized as follows. In the following section we give some definitions. In Section 3 we state the two selection algorithms and prove their performance theoretically. In the last section we compare and analyze them experimentally.

## 2 Preliminaries

Throughout this paper, we use  $\mathcal{H}$  and  $n$  to denote the set of hypotheses and its size, and use  $\mathcal{D}$  to denote a distribution on instances. We assume some  $EX_{\mathcal{D}}()$  that generates instances according to the distribution  $\mathcal{D}$ , and each selection algorithm can make use of  $EX_{\mathcal{D}}()$ . For any  $h \in \mathcal{H}$ , let  $\text{prc}_{\mathcal{D}}(h)$  denote the accuracy of  $h$ , that is, the probability that  $h$  gives a correct prediction to  $x$  for a randomly given  $x$  under the distribution  $\mathcal{D}$ . Let  $h_0$  denote the best hypothesis in  $\mathcal{H}$  (w.r.t.  $\mathcal{D}$ ); that is,  $\text{prc}_{\mathcal{D}}(h_0) = \max\{\text{prc}_{\mathcal{D}}(h) | h \in \mathcal{H}\}$ . Let  $\gamma_0$  denote  $\text{prc}_{\mathcal{D}}(h_0) - 1/2$ ; that is,  $\text{prc}_{\mathcal{D}}(h_0) = 1/2 + \gamma_0$ .

We use  $\rho_{\text{upper}}$  and  $\rho_{\text{lower}}$  to denote upper and lower tail probabilities of independent

Bernoulli trials. More specifically, for any  $t \geq 1$  and  $p$ ,  $0 \leq p \leq 1$ , consider  $t$  independent random variables  $X_1, \dots, X_t$  each of which takes 0 and 1 with probability  $1 - p$  and  $p$ . Then for any  $\varepsilon > 0$ , we define  $\rho_{\text{upper}}(p, \varepsilon, t)$  and  $\rho_{\text{lower}}(p, \varepsilon, t)$  as follows:

$$\rho_{\text{upper}}(p, \varepsilon, t) = \Pr\left\{\sum_{i=1}^t X_i > pt + \varepsilon t\right\}, \quad \text{and} \quad \rho_{\text{lower}}(p, \varepsilon, t) = \Pr\left\{\sum_{i=1}^t X_i < pt - \varepsilon t\right\}.$$

For these tail probabilities, several bounds have been used in the literature; here we make use of the following ones (see, e.g., [6]).

**Theorem 2.1.** (Hoeffding bound)

For some constant  $c_H > 0$ , and for any  $p$ ,  $\varepsilon$ , and  $t$ , we have

$$\rho_{\text{upper}}(p, \varepsilon, t) < \exp(-c_H \varepsilon^2 t), \quad \text{and} \quad \rho_{\text{lower}}(p, \varepsilon, t) < \exp(-c_H \varepsilon^2 t).$$

**Remark.** The Hoeffding bound used in the literature uses  $c_H = 2$ . Later in this paper, we will use different constants that work respectively in a certain situation.

By using this bound, we can estimate the sufficient number of examples to guarantee that Batch Selection, the simple hypothesis selection algorithm, yields a hypothesis of reasonable accuracy with high probability. (In the following, we use  $\text{BS}(\delta, \gamma, m)$  to denote the execution of Batch Selection for parameters  $\delta$ ,  $\gamma$  and  $m$ , the sample size. Recall that the hypotheses space, its size, and the accuracy of best hypothesis is fixed, throughout this paper, to  $\mathcal{H}$ ,  $n$ , and  $1/2 + \gamma_0$ .)

**Theorem 2.2.** For any  $\gamma$  and  $\delta$ ,  $0 < \gamma, \delta < 1$ , if  $\gamma \leq \gamma_0$  and  $m = 16 \ln(2n/\delta)/(c_H \gamma^2)$  then with probability more than  $1 - \delta$ ,  $\text{BS}(\gamma, \delta, m)$  yields some hypothesis  $h$  with  $\text{prc}_{\mathcal{D}}(h) \geq 1/2 + \gamma_0/2$ .

**Proof.** Follows from the Hoeffding bound in Theorem 2.1.  $\square$

### 3 On-line Selection Algorithms and Their Analysis

Here we present our two on-line selection algorithms and investigate their reliability and efficiency theoretically. In our analysis of the algorithms we count each while-iteration as one step; thus, the number of steps is equal to the number of examples needed in the algorithm. By “at the  $t$  step” we precisely mean “at the point just after the  $t$ th while-iteration.” Throughout this section, we denote by  $\#_t(h)$  the number of examples for which the hypothesis  $h$  succeeds within  $t$  steps. It will be also useful for our analysis to partition the hypothesis space in two sets depending on the precision of each hypothesis. Thus, let

$\mathcal{H}_{\text{good}}$  (resp.,  $\mathcal{H}_{\text{bad}}$ ) denote the set of hypotheses  $h$  such that  $\text{prc}_{\mathcal{D}}(h) \geq 1/2 + \gamma_0/2$  (resp.,  $\text{prc}_{\mathcal{D}}(h) < 1/2 + \gamma_0/2$ ). This partition can be done in an arbitrary way. The complexity of our algorithms depends on it but can be easily adapted to a more restrictive condition (for instance  $h \in \mathcal{H}_{\text{good}}$  if  $\text{prc}_{\mathcal{D}}(h) \geq 1/2 + 3\gamma_0/4$ ) if it is needed for a particular application. Obviously, the more demanding is the definition of  $\mathcal{H}_{\text{good}}$ , the greater is the complexity of our algorithms.

In our analysis, we ignore small difference occurring by taking ceiling or floor function, or by computing real number with finite precision.

### 3.1 Constrained Selection Algorithm

We begin by introducing a function that is used to determine an important parameter of our algorithm. For a given  $n$ ,  $\delta$ , and  $\gamma$ , define  $b_{\text{CS}}(n, \delta, \gamma)$  by

$$b_{\text{CS}}(n, \delta, \gamma) = \frac{16}{c_{\text{H}}\gamma^2} \cdot \ln \left( \left( \frac{2n}{\delta} \right) \left( \frac{16e}{c_{\text{H}}(e-1)\gamma^2} \right) \right) = \frac{16}{c_{\text{H}}\gamma^2} \cdot \ln \left( \frac{32en}{c_{\text{H}}(e-1)\delta\gamma^2} \right).$$

Now our first algorithm, that we denote by CS from constrained selection, is stated as follows.

**Algorithm** CS( $\delta, \gamma$ )

```

 $B \leftarrow 3\gamma b_{\text{CS}}(n, \delta, \gamma)/4;$ 
set  $w(h) \leftarrow 0$  for all  $h \in \mathcal{H}$ ;
while  $\forall h \in \mathcal{H}$  [ $w(h) < B$ ] do
     $(x, b) \leftarrow EX_{\mathcal{D}}();$ 
     $\mathcal{H}' \leftarrow \{ h \in \mathcal{H} : h(x) = b \}; n' \leftarrow |\mathcal{H}'|;$ 
    for each  $h \in \mathcal{H}$  do
        if  $h \in \mathcal{H}'$  then  $w(h) \leftarrow w(h) + 1 - n'/n;$ 
        else  $w(h) \leftarrow w(h) - n'/n;$ 
    end-for
end-while
output  $h \in \mathcal{H}$  with the largest  $w(h)$ ;
```

Note that the number  $n'$  of successful hypotheses may vary at each step, which makes our analysis difficult. For avoiding this difficulty, we approximate  $n'$  as  $n/2$ ; that is, we assume that a half of hypotheses in  $\mathcal{H}$  always succeeds on a given example. In other words, we assume the following.

**Assumption.** After  $t$  steps (i.e., after  $t$  while-iterations), the following holds for each  $h \in \mathcal{H}$ .

$$w(h) = \#_t(h) - t/2,$$

**Remark.** In fact, we can modify CS to the one satisfying this assumption; that is, use a fixed, i.e.,  $1/2$ , decrement term instead of  $n'/n$ . As our experiments show, both algorithms seem to have almost the same reliability, while the modified algorithm has more stable complexity. We believe, however, that the original algorithm is more efficient in many practical applications. (See the next section for our experiments and discussion.)

First we investigate the reliability of this algorithm.

**Theorem 3.1.** For any  $\gamma$  and  $\delta$ ,  $0 < \gamma, \delta < 1$ , if  $\gamma \leq \gamma_0$ , then with probability more than  $1 - \delta$ ,  $\text{CS}(\gamma, \delta)$  yields some hypothesis  $h \in \mathcal{H}_{\text{good}}$ .

**Proof.** We estimate the error probability  $P_{\text{err}}$ , i.e., the probability that CS chooses some hypothesis with  $\text{prc}_{\mathcal{D}}(h) < 1/2 + \gamma_0/2$ , and show that it is less than  $\delta$ , in the following way.

$$\begin{aligned} P_{\text{err}} &= \Pr_{\text{CS}} \left\{ \bigcup_{t \geq 1} [\text{CS stops at the } t\text{th step and yields some } h \in \mathcal{H}_{\text{bad}}] \right\} \\ &\leq \Pr_{\text{CS}} \left\{ \bigcup_{t \geq 1} [\exists h \in \mathcal{H}_{\text{bad}} [w(h) \text{ reaches } B \text{ at the } t\text{th step (for the first time)}] \right. \\ &\quad \left. \wedge \forall h \in \mathcal{H}_{\text{good}} [w(h) \text{ has not reached } B \text{ within } t-1 \text{ steps}]] \right\} \\ &\leq \sum_{h \in \mathcal{H}_{\text{bad}}} \Pr_{\text{CS}} \left\{ \bigcup_{t \geq 1} [[w(h) \text{ reaches } B \text{ within } t \text{ steps}] \right. \\ &\quad \left. \wedge [w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps}]] \right\}. \end{aligned}$$

Let  $\tilde{t}_0 = b_{\text{CS}}(n, \delta, \gamma)$  and  $t_0 = (\gamma/\gamma_0)\tilde{t}_0$ . (Note that  $t_0 \leq \tilde{t}_0$ .) We estimate the above probability considering two cases:  $t \leq t_0$  and  $t \geq t_0 + 1$ . That is, we consider the following two probabilities.

$$\begin{aligned} P_1(h) &= \Pr_{\text{CS}} \left\{ \bigcup_{t \leq t_0} [[w(h) \text{ reaches } B \text{ within } t \text{ steps}] \right. \\ &\quad \left. \wedge [w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps}]] \right\}, \text{ and} \\ P_2(h) &= \Pr_{\text{CS}} \left\{ \bigcup_{t_0+1 \leq t} [[w(h) \text{ reaches } B \text{ within } t \text{ steps}] \right. \\ &\quad \left. \wedge [w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps}]] \right\}. \end{aligned}$$

In Lemma 3.2 and Lemma 3.3 below, we prove that both  $P_1(h)$  and  $P_2(h)$  are bounded by  $\delta/2n$  for any  $h \in \mathcal{H}_{\text{bad}}$ . Therefore we have

$$P_{\text{err}} \leq \sum_{h \in \mathcal{H}_{\text{bad}}} P_1(h) + P_2(h) \leq n \left( \frac{\delta}{2n} + \frac{\delta}{2n} \right) = \delta.$$

□

**Lemma 3.2.** For any  $h \in \mathcal{H}_{\text{bad}}$ , we have  $P_1(h) \leq \delta/2n$ .

**Proof.** We bound the probability  $P'_1(h) = \Pr_{\text{CS}}\{\cup_{t \leq t_0} [w(h) \text{ reaches to } B \text{ within } t \text{ steps}]\}$ . Clearly  $P_1(h) \leq P'_1(h)$ .

The probability  $P'_1(h)$  is in fact the same as the probability that  $w(h)$  reaches to  $B$  in  $t_0$  steps. Now suppose that  $w(h)$  reaches to  $B$  in  $t_0$  steps. Then for some  $t \leq t_0$ ,  $w(h) \geq B$  at the  $t$ th step (i.e., just after the  $t$ th step). From our assumption, we have  $w(h) = \#_t(h) - t/2$  at the  $t$ th step. Also recall that  $B = 3\gamma\tilde{t}_0/4$  and that  $E[\#_t(h)] < t/2 + \gamma_0 t/2$  (since  $h \in \mathcal{H}_{\text{bad}}$ ). Hence,

$$\begin{aligned} & w(h) \geq B \text{ at the } t \text{th step} \\ \Leftrightarrow & \#_t(h) - t/2 \geq B = 3\gamma\tilde{t}_0/4 = 3\gamma_0 t_0/4 \\ \Leftrightarrow & \#_t(h) \geq E[\#_t(h)] + (t/2 + 3\gamma_0 t_0/4 - E[\#_t(h)]) \\ \Rightarrow & \#_t(h) > E[\#_t(h)] + (3\gamma_0 t_0/4 - \gamma_0 t/2) > E[\#_t(h)] + \gamma_0 t_0/4. \end{aligned}$$

Therefore, if  $w(h)$  reaches to  $B$  within  $t_0$  steps, then  $\#_t(h) > E[\#_t(h)] + \gamma_0 t_0/4$  for some  $t \leq t_0$ . Hence, by using the Hoeffding bound 2.1, we can derive the following bound. (Here recall that  $\gamma \leq \gamma_0$  and  $t_0 = (\gamma/\gamma_0)\tilde{t}_0$ .)

$$P'_1(h) \leq \exp\left(-c_H\left(\frac{\gamma_0 t_0}{4t}\right)^2 t\right) \leq \exp\left(-\frac{c_H \gamma_0^2 t_0}{16}\right) \leq \exp\left(-\frac{c_H \gamma^2 \tilde{t}_0}{16}\right).$$

On the other hand, by our choice of  $\tilde{t}_0$  (i.e.,  $b_{\text{CS}}$ ), we have  $\exp(-c_H \gamma^2 \tilde{t}_0/16) < \delta/2n$ .  $\square$

**Lemma 3.3.**  $P_2(h) \leq \delta/2n$ .

**Proof.** First we note the following.

$$\begin{aligned} P_2(h) &= \Pr_{\text{CS}}\left\{\bigcup_{t_0+1 \leq t} [\{w(h) \text{ reaches to } B \text{ within } t \text{ steps}\} \wedge \{w(h_0) \text{ has not reached to } B \text{ within } t-1 \text{ steps}\}]\right\} \\ &\leq \Pr_{\text{CS}}\left\{\bigcup_{t_0+1 \leq t} [w(h_0) \text{ has not reached to } B \text{ within } t-1 \text{ steps}]\right\} \\ &\leq \sum_{t_0+1 \leq t} \Pr_{\text{CS}}\{w(h_0) \text{ has not reached to } B \text{ within } t-1 \text{ steps}\}. \end{aligned}$$

Thus, we estimate the probability  $P'_2(h, t) = \Pr_{\text{CS}}\{w(h_0) \text{ has not reached to } B \text{ in } t \text{ steps}\}$ , for each  $t \geq t_0$ .

Here we modify CS slightly (which we call CS') so that it does not terminate even if some of the weights reaches to  $B$ , and let  $w_t(h)$  denote the weight of  $h$  at the  $t$ th step in the execution of CS'. Note that if  $w(h_0)$  has not reached to  $B$  in CS within  $t$  steps (including the  $t$ th step), then  $w_t(h_0) < B$  in CS'. On the other hand, we have

$$\begin{aligned}
w_t(h_0) < B &\Leftrightarrow \#_t(h_0) - t/2 < B = 3\gamma\tilde{t}_0/4 = 3\gamma_0t_0/4 \\
&\Leftrightarrow \#_t(h_0) < \mathbb{E}[\#_t(h_0)] + (t/2 + 3\gamma_0t_0/4 - \mathbb{E}[\#_t(h_0)]) \\
&\Leftrightarrow \#_t(h_0) < \mathbb{E}[\#_t(h_0)] + (3\gamma_0t_0/4 - \gamma_0t) \leq \mathbb{E}[\#_t(h)] - \gamma_0t/4.
\end{aligned}$$

Therefore, if  $w(h_0)$  has not reached to  $B$  in  $t$  steps in CS, then  $\#_t(h_0) < \mathbb{E}[\#_t(h)] - \gamma_0t/4$  in CS'. Hence, by using the Hoeffding bound again, we get  $P'_2(h, t) < \exp(-c_H\gamma_0^2t/16)$ .

Now we estimate  $\sum_{t_0+1 \leq t} P'_2(h, t)$ . First for any  $\Delta \geq 0$ , consider  $P'_2(h, t_0 + \Delta)$ . From the above, we have  $P'_2(h, t) < P_0 \cdot \exp(-(c_H\gamma_0^2/16)\Delta)$ , where  $P_0 = \exp(-(c_H\gamma_0^2t_0/16))$ . Hence, if  $\Delta \geq 16/c_H\gamma_0^2$ , then  $P_1(h, t_0 + \Delta) < P_0 \cdot e^{-1}$ . In general, if  $\Delta \geq k(16/c_H\gamma_0^2)$ , then  $P_1(t_0 + \Delta) < P_0 \cdot e^{-k}$ . Therefore we have<sup>1</sup>

$$\begin{aligned}
\sum_{t \geq t_0} P'_2(h, t) &= \sum_{\Delta \geq 0} P'_2(h, t_0 + \Delta) \\
&\leq P_0 \cdot \frac{16}{c_H\gamma_0^2} \cdot \frac{1}{1 - e^{-1}} < \frac{\delta}{2n} \cdot \frac{c_H(e-1)\gamma^2}{16e} \cdot \frac{16e}{c_H(e-1)\gamma_0^2} \leq \frac{\delta}{2n}.
\end{aligned}$$

(Note that  $P_0 \leq \exp(-(c_H\gamma^2\tilde{t}_0/16))$ , which is less than  $(\delta/2n)(c_H(e-1)\gamma^2/16e)$  by our choice of  $\tilde{t}_0$  (i.e.,  $b_{\text{CS}}$ ).)  $\square$

Though valid, our estimation of error probability is not tight, and it may not give us a useful bound  $B$  for practical applications. Here under a certain assumption (i.e., the independence of hypotheses), we can derive a much better formula for computing  $B$ .

**Theorem 3.4.** Consider a modification of CS, where we use the following definition for  $b_{\text{CS}}$ .

$$b_{\text{CS}}(n, \delta, \gamma) = \frac{16 \ln(2n/\delta)}{c_H\gamma^2}.$$

Assume that for any  $h$  and  $h'$ , the correctness of  $h$  on a randomly given example  $x$  is independent from that of  $h'$ . (See the proof below for the precise condition.) Then we can show the same reliability for CS as Theorem 3.1 for the modified algorithm.

**Proof.** It is easy to see that the new  $b_{\text{CS}}$  is good enough for showing Lemma 3.2 (i.e.,  $P_1(h) \leq \delta/2n$ ); on the other hand, the proof of Lemma 3.3 requires the previous  $b_{\text{CS}}$ . Thus, we do over the estimation of  $P_2(h)$  again.

This time we bound  $P_2$  as follows.

---

<sup>1</sup>Precisely speaking, the factor  $8/\gamma_0^2$  should be  $\lceil 8/\gamma_0^2 \rceil$ ; but the effect of the ceiling function is negligible, we omit it for simplifying our discussion.

$$\begin{aligned}
P_2(h) &= \Pr_{\text{CS}} \left\{ \bigcup_{t_0+1 \leq t} \left[ [w(h) \text{ reaches } B \text{ within } t \text{ steps}] \right. \right. \\
&\quad \left. \left. \wedge [w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps}] \right] \right\} \\
&\leq \sum_{t_0+1 \leq t} \Pr_{\text{CS}} \left\{ [w(h) \text{ reaches } B \text{ within } t \text{ steps}] \right. \\
&\quad \left. \wedge [w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps}] \right\}
\end{aligned}$$

Now we use our assumption, the independence of hypotheses; more specifically, we assume, for any  $h \in \mathcal{H}_{\text{bad}}$ , that  $\Pr \{ [w(h) \text{ reaches } B \text{ within } t \text{ steps}] \wedge [w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps}] \} = \Pr \{ [w(h) \text{ reaches } B \text{ within } t \text{ steps}] \} \times \Pr \{ [w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps}] \}$ . Then from the above, we obtain the following bound.

$$\begin{aligned}
P_2(h) &\leq \sum_{t_0+1 \leq t} \Pr_{\text{CS}} \{ w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps} \} \\
&\quad \times \Pr_{\text{CS}} \{ w(h) \text{ reaches } B \text{ within } t \text{ steps} \}.
\end{aligned}$$

On the other hand, we can show that, for any  $t \geq t_0 + 1$ ,  $\Pr_{\text{CS}} \{ w(h_0) \text{ has not reached } B \text{ within } t-1 \text{ steps} \} \leq \delta/2n$ . (See the proof of Lemma 3.3.) Therefore, we have

$$P_2(h) \leq \sum_{t_0+1 \leq t} \frac{\delta}{2n} \times \Pr_{\text{CS}} \{ w(h) \text{ reaches } B \text{ within } t \text{ steps} \} \leq \frac{\delta}{2n}.$$

□

It may be unlikely that  $h_0$  is independent from *all* hypotheses in  $\mathcal{H}_{\text{bad}}$ . We may reasonably assume, however, that for any  $h \in \mathcal{H}_{\text{bad}}$ , there exists some  $h' \in \mathcal{H}_{\text{good}}$  such that  $h$  and  $h'$  are (approximately) independent, and our proof above works similarly for such an assumption. Thus, in most cases, we may safely use the simplified version of  $b_{\text{CS}}$ , and we will use it in the following discussion.

Next let us discuss the complexity of our algorithm CS. Here by “complexity”, we mean the number of steps that  $\text{CS}(\delta, \gamma)$  needs to yield a hypothesis, or in other words, the number of examples used to select a hypothesis.

Consider the execution of CS on some  $\delta > 0$  and  $\gamma \leq \gamma_0$ . It is easy to see that, after  $t$  steps, the weight of  $h_0$  becomes  $\gamma_0 t$  on average. Thus, on average, the weight reaches  $B$  in  $B/\gamma_0$  steps<sup>2</sup>. From this observation, we may use the following function for the average complexity of  $\text{CS}(\delta, \gamma)$ .

$$t_{\text{CS}}(n, \delta, \gamma, \gamma_0) = \frac{B}{\gamma_0} = \frac{12 \ln(2n/\delta)}{c_H \gamma \gamma_0}.$$

---

<sup>2</sup>Precisely speaking, our argument is not mathematically correct, because we estimate here  $\min\{t | \mathbb{E}[w_t(h_0)] \geq B\}$ , whereas what we need to estimate is  $\mathbb{E}[\min\{t | w_t(h_0) \geq B\}]$ .

## 3.2 Adaptive Selection Algorithm

In this section we give a different algorithm that does not use any knowledge on the accuracy of the best hypothesis in the class ( recall that algorithm CS used the knowledge of a lower bound on  $\gamma_0$ .). To achieve this goal, we modify the condition of the while loop so it is changing adaptively according to the number of examples we are collecting. We call the algorithm *AS* from adaptive selection. The algorithm is stated as follows.

**Algorithm** AS( $\delta$ )

```

 $S \leftarrow \emptyset; t \leftarrow 0; \epsilon \leftarrow 1/5;$ 
while  $\forall h \in \mathcal{H} [\#_t(h) \leq t/2 + 5t\epsilon/2]$  do
     $(x, b) \leftarrow EX_{\mathcal{D}}();$ 
     $S \leftarrow S \cup \{(x, b)\}; t \leftarrow t + 1;$ 
     $\epsilon \leftarrow \sqrt{4 \ln(3n/\delta)/(c_H t)};$ 
end-while
output  $h \in \mathcal{H}$  with the largest  $\#_t(h)$ ;

```

**Remark.** The condition of the while-loop is trivially satisfied until the algorithm collects enough number of examples for  $S$ , i.e.,  $\|S\| > 4 \ln(3n/\delta)/(c_H(1/5)^2)$ . Thus, in practice, we start the while-loop after obtaining  $4 \ln(3n/\delta)/(c_H(1/5)^2)$  examples for  $S$ .

Again we begin by investigating the reliability of this algorithm.

**Theorem 3.5.** For any  $\delta$ ,  $0 < \delta < 1$ , with probability more than  $1 - \delta$ , AS( $\delta$ ) yields some hypothesis  $h \in \mathcal{H}_{\text{good}}$ .

**Proof.** Our goal is to show that when the algorithm stops it outputs a hypothesis  $h \in \mathcal{H}_{\text{good}}$  with probability more than  $1 - \delta$ . That is, we want to show the following probability is larger than  $1 - \delta$ .

$$\begin{aligned}
P_{\text{crct}} &= \Pr_{\text{AS}} \{ \bigcup_{t \geq 1} [\text{AS stops at the } t\text{th step and yields some } h \in \mathcal{H}_{\text{good}}] \} \\
&= \sum_{t \geq 1} \Pr_{\text{AS}} \{ \text{AS stops at the } t\text{th step and yields some } h \in \mathcal{H}_{\text{good}} \} \\
&= \sum_{t \geq 1} \Pr_{\text{AS}} \{ \text{AS yields some } h \in \mathcal{H}_{\text{good}} \mid \text{AS stops at the } t\text{th step} \} \\
&\quad \times \Pr_{\text{AS}} \{ \text{AS stops at the } t\text{th step} \}.
\end{aligned}$$

Consider any  $t \geq 1$ , and assume in the following that the algorithm stops at the  $t$ th step, i.e., just after the  $t$ th while-iteration. (Thus, we discuss here probability under the condition that AS stops at the  $t$ th step.) Let  $\epsilon_t$  and  $S_t$  be the value of  $\epsilon$  and  $S$  at the  $t$ th

step. Also let  $h$  be the hypothesis that AS yields; that is,  $\#_t(h)$  is the largest at the  $t$ th step.

By our choice of  $\varepsilon_t$ , we know that  $t = 4 \ln(3n/\delta)/(c_H \varepsilon_t^2)$ , and thus, by Lemma 3.6 given below, the following inequalities hold with probability  $> 1 - \delta$ .

$$\text{prc}_{\mathcal{D}}(h_0) \leq \text{prc}_{\mathcal{D}}(h) + \varepsilon_t, \quad \text{and} \quad |\text{prc}_{\mathcal{D}}(h) - \#_t(h)/t| \leq \varepsilon_t/2.$$

From the second inequality, we have that  $\#_t(h) \leq t(\varepsilon_t/2 + \text{prc}_{\mathcal{D}}(h))$ , and since we know that  $1/2 + \gamma_0 = \text{prc}_{\mathcal{D}}(h_0) \geq \text{prc}_{\mathcal{D}}(h)$ , we get that  $\#_t(h) \leq t/2 + t\gamma_0 + t\varepsilon_t/2$ . Moreover, since the algorithm stopped, the condition of the while-loop is not satisfied and thus, the following holds.

$$t/2 + 5t\varepsilon_t/2 \leq \#_t(h) \leq t/2 + t\gamma_0 + t\varepsilon_t/2$$

This implies that  $\varepsilon_t \leq \gamma_0/2$ . With this fact together with the first inequality above (i.e.,  $\text{prc}_{\mathcal{D}}(h_0) \leq \text{prc}_{\mathcal{D}}(h) + \varepsilon_t$ ), we can conclude that  $1/2 + \gamma_0/2 \leq \text{prc}_{\mathcal{D}}(h)$ .

Therefore, for any  $t \geq 1$ , we have  $\Pr_{\text{AS}}\{\text{AS yields some } h \in \mathcal{H}_{\text{good}} \mid \text{AS stops at the } t\text{th step}\} > 1 - \delta$ . This, together with the fact that  $\sum_{t \geq 1} \Pr_{\text{AS}}\{\text{AS stops at the } t\text{th step}\} = 1$  proves the theorem.  $\square$

**Lemma 3.6.** For a given  $\varepsilon$ ,  $0 < \varepsilon \leq 1$ , let  $t = 4 \ln(3n/\delta)/(c_H \varepsilon^2)$ , and consider the point in the execution of the algorithm just after the  $t$ th step. Then for any  $h \in \mathcal{H}$  such that  $\#_t(h) \geq \#_t(h_0)$  we have

$$\Pr_{\text{AS}}\{\text{prc}_{\mathcal{D}}(h_0) \leq \text{prc}_{\mathcal{D}}(h) + \varepsilon \wedge [|\text{prc}_{\mathcal{D}}(h) - \#_t(h)/t| \leq \varepsilon/2]\} > 1 - \delta.$$

**Proof.** Fix any  $h \in \mathcal{H}$  such that  $\#_t(h) \geq \#_t(h_0)$  and let  $A(h)$  and  $B(h)$  denote the following conditions.

$$\begin{aligned} A(h) &\Leftrightarrow [\text{prc}_{\mathcal{D}}(h_0) \leq \text{prc}_{\mathcal{D}}(h) + \varepsilon] \wedge [|\text{prc}_{\mathcal{D}}(h) - \#_t(h)/t| \leq \varepsilon/2], \quad \text{and} \\ B(h) &\Leftrightarrow [\text{prc}_{\mathcal{D}}(h_0) - \#_t(h_0)/t \leq \varepsilon/2] \wedge [|\text{prc}_{\mathcal{D}}(h) - \#_t(h)/t| \leq \varepsilon/2]. \end{aligned}$$

We first show that  $B(h)$  implies  $A(h)$ . Notice that  $B(h)$  implies that

$$[(\text{prc}_{\mathcal{D}}(h_0) - \#_t(h_0)/t) + (\#_t(h)/t - \text{prc}_{\mathcal{D}}(h)) \leq \varepsilon] \wedge [|\text{prc}_{\mathcal{D}}(h) - \#_t(h)/t| \leq \varepsilon/2].$$

Rewriting we obtain that

$$[(\#_t(h)/t - \#_t(h_0)/t) + (\text{prc}_{\mathcal{D}}(h_0) - \text{prc}_{\mathcal{D}}(h)) \leq \varepsilon] \wedge [|\text{prc}_{\mathcal{D}}(h) - \#_t(h)/t| \leq \varepsilon/2],$$

and since  $\#_t(h) \geq \#_t(h_0)$ , it must hold that

$$[\text{prc}_{\mathcal{D}}(h_0) \leq \text{prc}_{\mathcal{D}}(h) + \varepsilon] \wedge [|\text{prc}_{\mathcal{D}}(h) - \#_t(h)/t| \leq \varepsilon/2],$$

which is condition  $A(h)$ .

Now we show that  $\Pr_{\text{AS}}\{\neg B(h)\} < \delta$ . Thus, by the union bound and the Hoeffding bound (Theorem 2.1), the probability over the choice of sample  $S$  of size  $t$  (which is the same as the probability over the execution of AS until the  $t$ th step) that there exists one  $h \in \mathcal{H}$  such that  $B(h)$  does not hold is less than  $3n\exp(c_H(\varepsilon/2)^2 t)$ , which is, by choice of  $t$ , equal to  $\delta$ . Then since  $\neg A(h) \Rightarrow \neg B(h)$ , the lemma follows.  $\square$

Next we discuss the complexity of the algorithm. Here we can prove the following bound.

**Theorem 3.7.** For any  $\delta$ ,  $0 < \delta < 1$ , with probability more than  $1 - \delta$ ,  $\text{AS}(\delta)$  terminates within  $64 \ln(3n/\delta)/c_H \gamma_0^2$  steps.

**Proof.** Here we use the same notation as above. Notice first that while we are in the while-loop, the value of  $\varepsilon$  is always strictly decreasing. Suppose that at some step  $t$ ,  $\varepsilon_t$  has became small enough so that  $4\varepsilon_t < \gamma_0$ . Then from Lemma 3.6 (the condition of the lemma always holds due to our choice of  $\varepsilon$ ), with probability  $> 1 - \delta$ , we have that  $t(\text{prc}_{\mathcal{D}}(h) - \varepsilon_t/2) \leq \#_t(h)$ , and  $\text{prc}_{\mathcal{D}}(h_0) - \varepsilon_t \leq \text{prc}_{\mathcal{D}}(h)$ . Putting these two inequalities together, we obtain that  $t/2 + t\gamma_0 - t\varepsilon_t - t\varepsilon_t/2 \leq \#_t(h)$  (since  $\text{prc}_{\mathcal{D}}(h_0) = 1/2 + \gamma_0$ ). Since we assumed that  $4\varepsilon_t < \gamma_0$ , we can conclude that  $t/2 + 5t\varepsilon_t/2 < \#_t(h)$ , and thus, the condition of the loop is falsified. That is, the algorithm terminates (at least) after the  $t$ th while-iteration.

Recall that  $\varepsilon_t$  is defined to be  $\sqrt{4 \ln(3|\mathcal{H}|/\delta)/(c_H t)}$  at any step. Thus, when we reach to the  $t$ th step with  $t = 64 \ln(3|\mathcal{H}|/\delta)/(c_H \gamma_0^2)$ , then it must hold that  $\varepsilon_t < \gamma/4$ , and by the above argument, the algorithm terminates with probability larger than  $1 - \delta$ .

**Remark.** Thus, we use the following function for our theoretical bound for the number of examples used by AS.

$$t_{\text{AS}}(n, \delta, \gamma) = \frac{16 \ln(3|\mathcal{H}|/\delta)}{c_H \gamma^2}.$$

$\square$

Again this theoretical bound is not tight. As we will see in the next section, our experiments show that the value of  $\varepsilon$ , when the algorithm stops, is close to  $\gamma_0/2$  instead of  $\gamma_0/4$ . Thus, the number of examples is much smaller than this theoretical bound.

## 4 Experimental Evaluation of the Algorithms

We first summarize three selection algorithms considered, and state functions that bound the sufficient number of examples to guarantee, in theory, that the algorithm selects with probability  $> 1 - \delta$  a hypothesis  $h$  with  $\text{prc}_{\mathcal{D}}(h) \geq 1/2 + \gamma_0/2$ . (Recall that we assume that a given hypothesis set  $\mathcal{H}$  has some  $h$  with  $\text{prc}_{\mathcal{D}}(h) \geq 1/2 + \gamma_0/2$ .)

- Batch Selection:  $\text{BS}(n, \delta, \gamma)$  (see Introduction)  
Bound:  $t_{\text{BS}}(n, \delta, \gamma) = 16 \ln(2n/\delta)/(c_H \gamma^2)$  on worst case. Condition:  $\gamma \leq \gamma_0$ .
- Constrained Selection:  $\text{CS}(n, \delta, \gamma)$   
Bound:  $t_{\text{CS}}(n, \delta, \gamma) = 12 \ln(2n/\delta)/(c_H \gamma \gamma_0)$  on average. Condition:  $\gamma \leq \gamma_0$ .
- Adaptive Selection:  $\text{AS}(n, \delta)$   
Bound:  $t_{\text{AS}}(n, \delta) = 64 \ln(3n/\delta)/(c_H \gamma_0^2)$  on worst case. Condition: None.

Thus, for example, if we know  $\gamma_0$  and use it as  $\gamma$ , then  $t_{\text{BS}}(n, \delta, \gamma)$  examples are enough to guarantee  $1 - \delta$  confidence for BS. We compare these theoretical bounds with the numbers that we obtained through experiments.

First we describe the setup used in our experiments. We decided to use synthetic data instead of real datasets so that we can investigate our algorithms in a wider range of parameter values. (In future work we are planning to evaluate also them with real data.)

The common fixed parameters involved in our experiments are  $\delta$ , the confidence parameter, and  $n$ , the number of hypotheses in  $\mathcal{H}$ . Notice that these two parameters are inside a logarithm in the above bounds; thus, results are not really affected by modifying them.

In fact, we verified this experimentally, and based on those results we set them to 18 for  $n$ , and 0.01 for  $\delta$ ; that is, we require confidence of 99%. The other parameter is the accuracy of the best hypothesis, which is specified by  $\gamma_0$ . In our experiments the value of  $\gamma_0$  ranges from 0.04 to 0.3 with a increment of 0.01 (that is, the accuracy of the best hypothesis ranges from 54% to 80% with a increment of 0.4%) and we have a total of 65 different values. For each  $\gamma_0$ , we distributed the 18 hypotheses in 9 groups of 2 hypotheses, where the accuracy of hypotheses in each group is set  $1/2 - \gamma$ ,  $1/2 - 3\gamma/4$ , ...,  $1/2 + 3\gamma/4$ ,  $1/2 + \gamma$ . The choice of the distribution of hypotheses accuracy does not affect the performance of neither BS nor AS (because their performance depends only on the accuracy of the best hypothesis). On the other hand, it seems to affect the performance of CS. For this reason, we also tried other distributions of the hypotheses accuracy for CS. For a random number generator, we used one explained in [7].

For each set of parameters, we generated a *success pattern* for each hypothesis  $h$ . A *success pattern* is a 0/1 string of 1000 bits that are used to determine whether the hypothesis  $h$  predicts correctly for a given example. That is, to simulate the behavior

of  $h$  on examples from  $EX_{\mathcal{D}}()$ , we just draw a random number  $i$  between 1 and 1000, and decide  $h$  predicts correctly/wrongly on the current example if the  $i$ th bit of the success pattern is 1/0. Finally, for every fixed setting of all the parameters, we run this experiments 30 times, i.e., run each algorithm 30 times, and averaged the results. This is what is reflected on the graphs we have throughout this section.

### 1. The Tightness of Theoretical Bounds

Let us assume that we know the value of  $\gamma_0$ , not just a lower bound. Then, from the bounds summarized first, one may think that, e.g., CS is more efficient than BS. It turned out, however, it is not the case. Our experiment shows that the number of required examples is similar among three algorithms, and the difference is the tightness of our theoretical bounds. Of course, this is for the case when  $\gamma_0$  is known, see the subsection below for a discussion on this issue.

We checked that the “necessary and sufficient” number of examples is proportional to  $1/\gamma_0^2$  (where  $n$  and  $\delta$  are fixed). Thus, we changed the parameter  $c_H$  to get the tightest bounds; that is, for each algorithm, we obtained the smallest  $c_H$  with which the algorithm does not make any mistake in 30 runs. The graph (a) of Figure 1 shows the number of examples needed by three algorithms with such almost optimal constants. There is not so much difference, in particular, between CS and AS. Thus, the tightness of our estimation seems to be the main factor of the difference of theoretical bounds when  $\gamma_0$  is known.

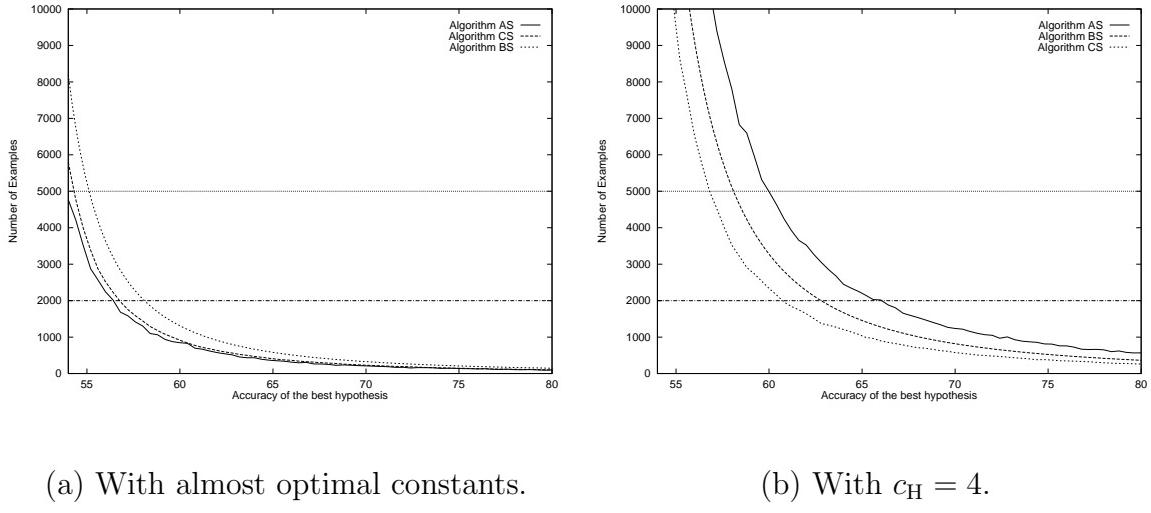


Figure 1: the number of examples vs.  $\gamma_0$

It is, however, impossible in real applications to estimate the optimal constant and get the tightest bound. Nevertheless, we can still get a better bound by a simple calculation. Recall that the Hoeffding bound is a general bound for tail probabilities of Bernoulli trials. While it may be hard to improve the constant  $c_H$  in general, we can numerically calculate a better one for a given set of parameters. For instance, for our experiments, we can safely

use  $c_H = 4$  instead of  $c_H = 2$ , and the difference is half; e.g.,  $t_{BS}(18, 0.01, 0.1)$  (so the best hypothesis has 60% of accuracy) is 6550 with  $c_H = 2$  but 3275 with  $c_H = 4$ . The graph (b) of Figure 1 shows the number of examples needed by three algorithms with  $c_H = 4$ . Thus, when using these algorithms, it is recommended to estimate first an appropriate constant  $c_H$ , and use it in the algorithms. For such usage, CS is the most efficient for the set of parameters we used.

## 2. Comparison of Three Algorithms

The graph (b) of Figure 1 indicates that CS is best (at least within this range of parameters) if  $\gamma_0$  or a good approximation of it is *known*. The situation differs a lot if we do not know  $\gamma_0$ . For example, if  $\gamma_0 = 0.2$  but it is underestimated as 0.05, then BS and CS need 13101 and 2308 examples, while AS needs only 1237 examples; thus, in that case AS is the most efficient. This phenomenon is shown in Figure 2, where we fixed  $\gamma_0$  to be 0.2% (so the accuracy of the best hypothesis is 70%), and we changed the value of the lower bound  $\gamma$  from 0.04 to 0.2. Algorithm AS is not affected by the value of  $\gamma$ , and hence it uses the same number of examples (the horizontal line in the graph). With this graph we can see that, for instance, when  $\gamma$  ranges from 0.04 to 0.058, algorithm AS is the most efficient, while from 0.058 to 0.2 algorithm CS becomes the best; but in any case, the difference is not so big within this range of  $\gamma$ . On the other hand, the performance of BS becomes considerably bad if we underestimate  $\gamma_0$  and the number of examples needed by this algorithm might become huge.

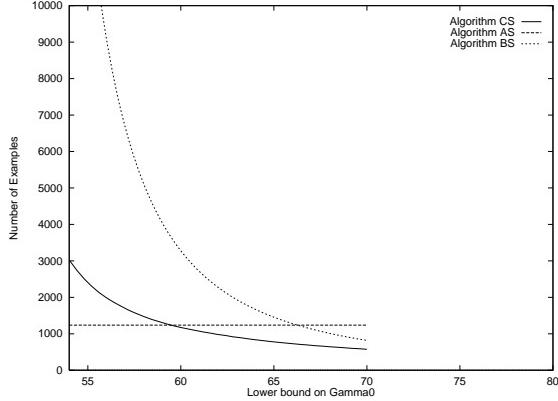


Figure 2:  $t$  vs.  $\gamma$  ( $\gamma_0 = 0.2$ )  
( $t$  denotes the number of examples.)

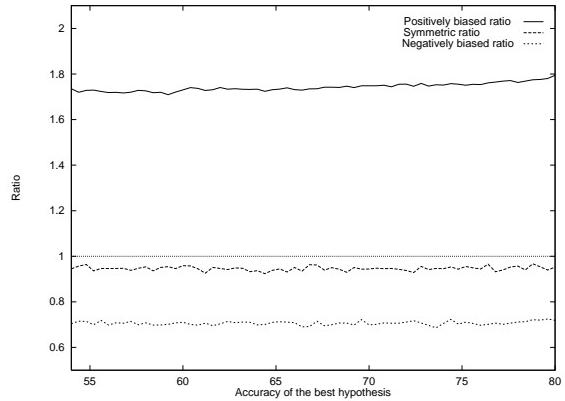


Figure 3:  $t/B\gamma_0$  vs.  $\gamma_0$

## 3. CS: Constant dec vs. Variable dec

For simplifying our theoretical analysis, we assumed that  $dec$  (recall that  $dec$  was  $n'/n$ ) is constant 1/2. In fact, there are two choices: either (i) to use constant  $dec$ , or (ii) to use

variable  $dec$ . We investigate whether it affects the performance of the algorithm CS. We verified that it does not affect at all the reliability of CS. On the other hand, it affects the efficiency of CS, i.e., the number of examples needed by CS.

Intuitively the following is clear: If the distribution of hypotheses accuracy is symmetric (like in the above experiment), then the number of successful hypotheses, at each step, is about  $n/2$ ; thus,  $dec \approx 1/2$ , and the number of examples does not change between (i) and (ii). On the other hand, if most of the hypotheses are better than  $1/2$  (resp., most of the hypotheses are worse than  $1/2$ ), then the number of examples gets larger (resp., smaller) in (ii) than in (i). We verified this intuition experimentally. Figure 3 shows the ratio between the number of examples and  $B/\gamma_0$  (which is always close to 1 if  $dec = 1/2$ ) for three different distributions of hypotheses accuracy: symmetric, positively biased, and negatively biased. Thus, when the distribution is negatively biased, which is the case in many applications, we recommend to use the original CS with variable  $dec$ .

#### 4. AS: $\varepsilon$ vs. $\gamma_0$ , and the Theoretical Bound

From the theoretical analysis of Theorem 3.7, we obtained that the algorithm stops with high probability when  $\varepsilon$  becomes smaller than  $\gamma_0/4$ . On the other hand, to guarantee the correctness of our algorithm (Theorem 3.5), we just need to conclude that  $\varepsilon$  is smaller than  $\gamma_0/2$ . This difference gets reflected in our theoretical bound for the number of examples. Our experiments (see Figure 4) showed that the number of examples is much smaller than the theoretical bound. The reason is that, in most cases, the algorithm stops much before  $\varepsilon$  becomes as low as  $\gamma_0/4$ ; it is more likely, that AS stops as soon as  $\varepsilon$  becomes slightly smaller than  $\gamma_0/2$ . Figure 5 reflect this phenomenon; the final value of  $\varepsilon$  is closer to  $\gamma_0/2$  than  $\gamma_0/4$ . (It is in fact on the  $\gamma/2.38$  line.) If we assume that the final value of  $\varepsilon$  is about  $\gamma_0/2.38$  then, by using the relation between  $t$  and  $\varepsilon$ , we can estimate the number of examples as  $4(2.38)^2 \ln(3n/\delta)/(c_H \gamma_0^2)$ .

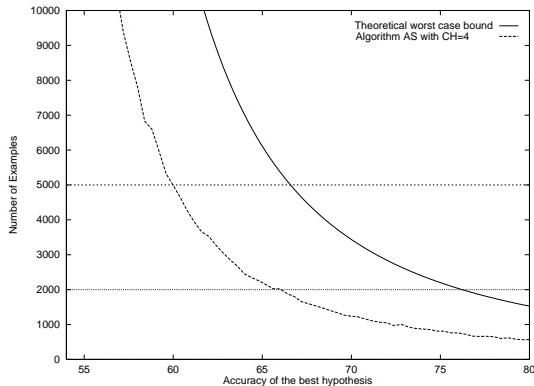


Figure 4:  $t$  and  $t_{AS}$

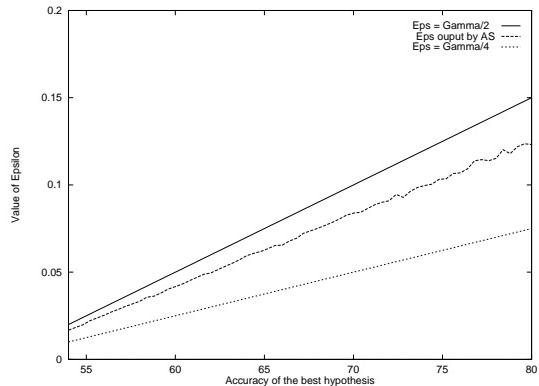


Figure 5: the final value of  $\varepsilon$

## References

- [1] P. Auer, R.C. Holte, and W. Mass. Theory and applications of agnostic PAC-learning with small decision trees. In *Proc. the 12th International Conference on Machine Learning*, 21-29, 1995.
- [2] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123-140, 1996.
- [3] T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error correcting ouput codes. *Journal of Artificial Intelligence Research*, 2:263–268, January 1995.
- [4] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. the Second European Conference on Computational Learning Theory*, Lecture Notes in Computer Science (subseries LNAI), 23-37, 1995. To appear in *Journal of Computer and System Science*.
- [5] R.C. Holte. Very simple classification rules perform well on most common datasets. *Machine Learning*, 11:63–91, 1993.
- [6] M.J. Kearns and U.V. Vazirani. *An Introduction to Computational Learning Theory*. Cambridge University Press, 1994.
- [7] R. Sedgewick, *Algorithms*, Addison-Wesley, second edition, 1998.
- [8] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134-1142, 1984.
- [9] S.M. Weiss, R.S. Galen and, P.V. Tadepalli. Maximizing the Predictive Value of Production Rules. *Artificial Intelligence*, 45, 47–71, 1990.